Learning Dimensions: Lessons from Field Studies

Conference or Workshop Item

Learning Dimensions: Lessons from Field Studies

Chris Martin
Life Sciences - CITR
University of Dundee
Dundee, UK
+44 1382 385828
c.j.martin@dundee.ac.uk

Janet Hughes
School of Computing and Communications
The Open University, UK
+44 131 549 7118
janet.hughes@open.ac.uk

John Richards
IBM T.J. Watson Research Center USA &
University of Dundee, UK
+1 914 945 2632
ajtr@us.ibm.com

ABSTRACT

In this paper, we describe work to investigate the creation of engaging programming learning experiences. Background research informed the design of four fieldwork studies involving a range of age groups to explore how programming tasks could best be framed to motivate learners. Our empirical findings from these four studies, described here, contributed to the design of a set of programming 'Learning Dimensions' (LDs). The LDs provide educators with insights to support key design decisions for the creation of engaging programming learning experiences. This paper describes the background to the identification of these LDs and how they could address the design and delivery of highly engaging programming learning tasks. A web application has been authored to support educators in the application of the LDs to their lesson design.

Keywords
Learning dimensions; motivation; programming.

1. INTRODUCTION

A substantial literature going back several decades (e.g. [25]) has explored various aspects of learning computer programming. [19] provide a detailed review and discussion of the literature pertaining to novice programmers. [22] review and discuss issues relating to development of CS1 courses. All of these authors note that programming is a multi-faceted task with many interrelated skills, and there is recognition that the transition from novice to expert is challenging. One result of this growing understanding is the major improvement in the educational technology developed to support learners [13]. Environments in themselves cannot provide the entirety of support for the different needs of novice learners, however. Educators also have awareness that whilst novices may apparently be making progress, their knowledge may be fragile and/or their lack of confidence can lead to 'stopper' behaviour. [20] suggest that knowledge in novice programmers is more complex than just 'knowing'. They describe the presence of 'fragile knowledge', which is categorised as missing, inert (learned but not used), or misplaced. They further observe that there can be different types of novice programmers: stoppers, movers and super movers [20].

A 'stopper' is characterised as person who is halted abruptly by an error or difficulty and does not have the inclination to tackle the problem independently. In contrast, a 'mover' is a learner with enthusiasm who views an error as a challenge rather than an obstacle. Perkins describes a third category of novice as a 'super mover': "tinkerers who are able to respond to errors but are unable to modify their program effectively and lose track of edits" [19]. Consequently, we judge that emotional response and enthusiasm are valid factors to consider in the process of learning to program. Our work takes an alternative approach to that of tool or IDE development, by considering the *context* in which learning takes place and the efficacy of learner motivation.

A recurring theme that emerges from the literature is that learning is fruitful in experiences that are personally meaningful for the learner. Aspects that increase personal motivation include personal, social, and contextual elements in addition to purely technical elements such as programming language and environment. Examples include the capacity to tap into and contribute to a community of like-minded learners, and the ability rapidly to make a thing that the learner values. The next section will introduce a set of field studies that explored learner-motivated programming.

2. FIELD STUDIES

Four field studies were performed with participants who were relative novices to programming, ranging in age from pre-school to university students. A range of qualitative and quantitative methods was used to gather data across the set of studies. An overview of each study is given next, together with the main conclusions reached from the observations and data collected.

2.1 Robot Dance

This study [15] was designed to explore how working with Arduino robots can support introductory programming learning. Arduino-based differential drive robots [1] were programmed for 1 to 2 hours by a total of 135 middle-school students (51% female), ranging in age from 12 to 16 years. The increased understanding of programming concepts by these learners, as measured by the difference in a pre- and post-test, was impressive: in a short space of time, learners were all able to use a textual language (C-style, rather than block-based) to create a simple program and demonstrably improve their knowledge in the areas of sequence, syntax and programming variables.

Further analysis suggested that a key aspect of this successful outcome was having a 'time to first task' of only around 10 minutes. This offered sufficient time to cover all the required knowledge to subsequently get a minimum viable robot program written and uploaded. Another important factor was the performance element of Robot Dance, enhanced as the workshops progressed by the inclusion of powerful external speakers, a wood-effect dance floor and a stage light served to increase the motivational effect of the end performance. Robot Dance delivered small pieces of skill and knowledge, giving the learners space to explore and experiment 'hands on' with the new material. The delivery of a new concept followed by space to explore the example was repeated several times. This cycle

ITiCSE '17, July 03-05, 2017, Bologna, Italy

ISBN 978-1-4503-4704-4/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3059009.3059046

supported a gradual increase in learner independence and task complexity. The next study reduced the degree of structure offered to learners.

2.2 Robot Dance in the Community

Here, the learners of introductory programming skills were given a greater degree of independence. Rather than a tight cycle of skill delivery and learning consolidation, learners were given a brief introduction and left to develop their Robot Dance, asking for assistance if and when they required it. The learning experience was organised to be drop-in, situated in a public shopping centre. Learners (members of the public) started at different times and could work as long as they wanted. Learners were also free to self-organise, which resulted in individuals, pairs, parent and child pairs, and larger groups. Following a brief introduction to Arduino, learners were given a very basic skeleton Arduino program to extend. To make this introduction concrete, learners were "walked-through" the program required to make the robot move forward a short distance. Once learners had successfully completed this task, the challenge of creating 20 seconds of dance moves was presented. The learners observed comprised a group of six parents and 35 children. Parents were considered where they performed an active role as opposed to passive observation. The children's ages ranged from five to 15 with the majority around seven. Four different groupings were observed: single child, child pairs, child parent pairs and multiple children and parents. All learners demonstrated an observable emotional response to the performance they had programmed. Learners exhibited pride in their creation, even though the audience was small. Observations from this study confirm that different learners require different degrees and types of support: freedom to experiment and self-direction worked well for some learners but was more challenging for others. It highlighted the extent to which programming has an emotional dimension. The next study investigated this further.

2.3 Whack-a-Mole

Robot Dance [15] demonstrated the extent to which the physical artefact mattered. The Whack-a-Mole study was designed to capture more insights into the emotions experienced by learners of programming, particularly when programming with different interfaces: a physical interface or a screen-based equivalent. The essence of the Whack-a-Mole game is simple: a stimulus occurs in one of several locations and the player reacts to it as quickly as possible. In the simplest version, a light comes on at random and stays on until the corresponding button is pressed. Using Arduino to give a physical interface, each of four LEDs has a corresponding button. When the light comes on, the player must press the corresponding physical button to progress through the game. The screen-based equivalent shows buttons on screen and the keyboard is used to press 'buttons'.

2.3.1 Capturing emotional responses

The Whack-a-Mole study involved 38 students (24% female) of a first level undergraduate computing module. In the first phase, learners were taught via three specific worked examples relating to programming with arrays and fixed loops. In the second phase, learners were required to demonstrate their understanding by applying the taught material to a novel problem. Learners were allocated at random into small practical groups of three or four. One set of groups used the physical Arduino interface whilst the other set used the screen-based equivalent. Emotional responses to programming were gathered at task completion by the Reflective Emotion Inventory (REI) derived from the HUMAINE project [22]. The REI questionnaire asks users to identify emotions they have experienced, to note the degree of intensity for each using a four-part Likert scale (0 indicated no emotion; 3 indicated that the emotion occurred intensely). They were asked also to offer some contextual information to describe why they experienced the given emotion. An example response is: annoyance, 3, "Getting the wires in the correct place".

2.3.2 Comparing responses to interfaces

When the physical set and screen-based sets of students' REI data responses were compared, the physical set was found to report greater intensity in all bar one of the emotional sub-categories (Figure 1). This matched the rich contextual data offered by the physical set. Where students worked with the physical artefact, they had a strongly positive experience.

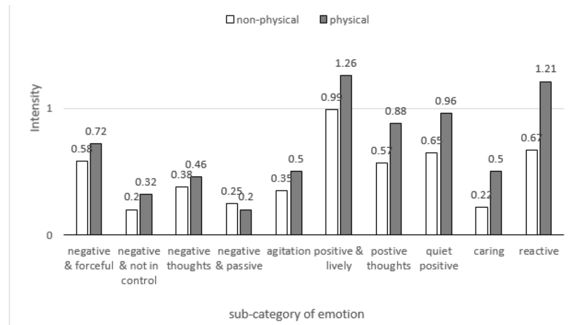

Figure 1: Whack-a-Mole emotional responses

Two of the positive emotions reported by the physical set were notably greater than that of the screen-based set: *positive & lively* and *reactive*. The Whack-a-Mole study uncovered a notable difference in emotional response to the learning experience for the students using a physical device compared to the students using a screen-based equivalent. Both sets of students described a range of negative emotions with similar levels of strength and for similar reasons, and a similar range of positive emotions. However, the physical set noted a greater strength of positive emotions associated with the learning experience. The next section describes a final field study, which was designed to further empower learners to create programs for problems they define and personally identify with.

2.4 Digital Makers

Additional aspects here were included to increase ownership, personalisation, and purpose. In previous studies, learners were tasked with solving challenges devised by the educator. Here, design decisions were less constrained for the learners, who could apply their newly acquired programming skills to solve a problem of their own. The study was part of One Day Digital, a series of digital making events for young people organised by [16]. Four events ran on consecutive weekends in different cities, engaging 48 young volunteer learners (17% female) from across the UK.

2.4.1 Programming set tasks with Arduino

In the morning, learners were walked through the process of wiring and programming some components with Arduino; for this stage learners worked as individuals. Following this, the programming of the component was demonstrated and then carried out by the learners. In three iterations of short demonstration followed by enactment by learners, three tasks were tackled: making an LED blink, using a potentiometer to control the blink rate and using a button to make the LED blink when pressed. To introduce a creative disruption to the flow of tuition, an idea-generation session was used to gather ideas posted together on a wall serving as an information radiator [24] for use later in the day. The learners were then guided through some additional Arduino output devices: servo, speaker and red green blue (RGB) LED. This gave the opportunity to show examples built into the Arduino IDE and the use of an external library for the servo. The final example they constructed was a red, green and blue colour mixer in which the colour of the LED was specified by three parameters passed to a user-defined function. Using the `random` function and bringing in sound (with loudspeakers playing beeps of a program-specified tone), the learners extended this to create a light and sound show.

2.4.2 Programming user-designed tasks

In the second stage, learners were given the chance to self-select groupings and build a physical app utilising the morning's teaching. Groups were given three hours to build a physical app based on one of the ideas they had selected from those they generated earlier. Before the workshop ended, participants were asked to complete the REI emotional response questionnaire.

The most striking result was the reporting of positive emotions as being far more intensely experienced than negative emotions (Figure 2). The physical apps session evoked a rich emotional response from the participants. Negative emotions experienced tied into problems reported in the literature about novice programming. Many of the error-prone features of coding match with those of physical prototyping, with bread boarding being particularly error-prone. Nonetheless, the minor irritations of an error-prone medium were outweighed by the strength of the positive emotions reported. Many positive emotions stemmed from a sense of overcoming challenges to produce something that worked.

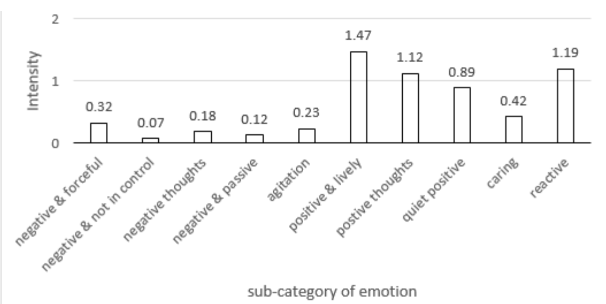

Figure 2: Digital Makers emotional responses

In summary, the Digital Makers study used ownership, personalisation, and purpose to create a highly engaging learning experience that resulted in strong positive emotional responses from learners. The next section describes how insights generated from these four studies were synthesised as a set of Learning Dimensions (LDs). The LDs follow the style that [9] proposed in their 'Cognitive Dimensions of Notations' framework, in which they outline a common vocabulary and reference point for the design and discussion of notations. It has served as a successful nucleus for a great deal of research relating to notations of many forms including code, sketching, algorithm visualization and musical staff notation. Cognitive Dimensions provided a common vocabulary that enabled researches to discuss insights. It is hoped that the LDs fulfil a similar role for educators making design decisions for motivating programming experiences.

3. LEARNING DIMENSIONS

3.1 Introduction

The aim of the Learning Dimensions is to provide a resource for computer science educators that can be used either in the design of new learning experiences or as a reflective toolkit for the review and improvement of existing learning experiences. The eight LDs address high-level aspects of learning experiences, particularly relating to practicalities of the design and delivery of a learning task. Each LD is described in its fullest form in [14], that is with (i) a detailed description; (ii) links to relevant literature; (iii) a summary of its rationale; (iv) examples from fieldwork; and (v) how it can be applied. Space constrains the description of each that can be provided here.

The first three dimensions relate to the design of activities to be particularly motivating and engaging. *Closed versus Open* describes the relative merits of designing learning tasks with or without a lot of detail and structure. *Cultural Relevance* describes the affordances presented by locating learning tasks within the learner's culture. *Recognition* describes opportunities arising from enabling learners to share their work. The next five LDs deal with the extent to which the programming experience can be learner-centred. *Space to Play* describes the impact of designing learning tasks that encourage iterative experimentation, for example with peers, and self-directed discovery of knowledge and skills. *Driver Shifting* describes the affordances of transferring the role of driving the learning experience from the educator to the learner, vice-versa, or via a collaboration of both. *Risk Reward* describes how the duration of tasks and the frequency of feedback can be adjusted to suit different learning experience needs. *Grouping* describes the possible arrangements of learners. *Session Shape* describes the affordance of the physical environment and how this may enhance or impede the learning experience.

3.2 Motivating programming

3.2.1 Closed versus open

This dimension encapsulates the extent to which activities have a well-defined structure, route, and end point. A good example of a closed problem is programming a robot to follow a line. There is little scope for the learner to take ownership. Towards the open end of the dimension would be a free choice activity where learners are able to demonstrate competency in a given skill through the creation of a piece of work that is not fully constrained by the educator. An example is creating a robot dance. In the fieldwork reported in section 2, Robot Dance in the Community and Digital Makers both exemplified the motivating effects of open programming tasks. Similar examples can be found in the literature. For instance, [21] conducted interviews with teams taking part in a RoboCup Junior (RCJ) event which invited teams of schoolchildren to compete using robots almost exclusively developed using LEGO Mindstorms [12]. Interviews were followed up by a detailed case study with one team. Two important factors that arose from the analysis were motivation and evidence of learning. One frequently reported reason for being motivated was the 'openness' of the task.

3.2.2 Cultural relevance

Often part of a learning experience involves creating a product of some kind, such as code or a sketch. The *Cultural Relevance* dimension considers where this product sits within the learner's culture. It prompts consideration of whether or not the tasks they are asked to perform are authentic and relevant to their daily life experience. If the learning experience is divorced from the world the learner inhabits, the cultural relevance will be low. Digital Makers, described in section 2, showed that ownership, personalisation, and purpose were related to strong positive emotional responses to programming. Personalisation and choice have been highlighted in the literature as important for increasing intrinsic motivation in learners. In one study of 72 fifth grade learners (10 to 11 year olds), [7] observed a powerful learning benefit in the personalised choice condition. Learners were observed to have not only increased motivation but also displayed a deeper engagement in the task.

3.2.3 Recognition

The *Recognition* dimension considers the potential for the learner to share the product of their learning. As early as nursery school, learners seek recognition from their teachers, peers, and parents. A good example of this is pleasure gained from the displaying of work on the walls of the learning environment for all to see. In section 2, *Recognition* was an important dimension in Robot Dance and Digital Makers when learners demonstrated their products in an end-of-workshop performance. *Recognition* is also possible via many educational programming tools which allow individuals to contribute to online communities of learners [7]. Learners can be inspired and informed by the work of others and in equal measure provide the inspiration and support for those who follow them. Considerable motivational affordances can come from sharing work and observing it being valued by others. [21] noted that a number of participants also identified placing the task in a social context as a factor contributing to motivation; the opportunity to share ideas and the pride associated with demonstrating expertise was reported to be important.

In LDs, the authors acknowledge that discussion is a richer mode of interaction than simply viewing work or broadcast-style presentation. With a discussion, a conversation about the product of the learning can take place between the learner and the audience. Learners gaining recognition through discussion are not just exposing a product, presenting an idea or artefact but they are also engaging in rich discourse about the artefact and process. This should ensure the audience and the learner reach a shared understanding of the idea or knowledge being presented. Where a deep interaction takes place, the learner's engagement and motivation will be affected by the amount of time, effort and interest the observers have invested in the interaction.

3.2.4 Space to play

The *Space to Play* dimension seeks to break down the traditional view of a teacher-learner relationship. It encapsulates the extent to which a learning experience offers and encourages learners to explore independently. *Space to Play* addresses the fact that space and independence may be intimidating for certain learners. It suggests a flexible structure to learning experiences, with frequent opportunities for learners to iterate over a concept that has just been introduced. This empowers individual learners to approach exploration on their own terms and take ownership of the learning experience. In [10], the teacher assumes the role of a facilitator rather than a gatekeeper to knowledge. This sets up a more progressive learning experience in which learners have a degree of influence on the direction of their learning.

In the Digital Maker study, *Space to Play* was integral to the design of the learning experience. The majority of the morning was spent learning about Arduino programing and electronics. Programming is a high precision error-prone activity; electronics prototyping has similar characteristics. To support this, the session was designed around frequent short *Spaces to Play*. A piece of programing and electronics was demonstrated and learners were given a time-boxed opportunity to try the task for themselves and experiment.

3.2.5 Driver Shift

This dimension attempts to capture who is driving the learning experience, i.e. controlling it at a given point in time. For example, a classic higher education style lecture where the lecturer projects content to the learners for a sustained period would have a low degree of *Driver Shift*. In contrast, a guided practical session with a tight cycle, in which learners are shown a brief example and then given space to try it, would have a high degree of *Driver Shift*. The concept emerged particularly from the Digital Makers study. As the day progressed and competence with newly acquired skills and knowledge grew, the length of the learner-driven blocks was increased and the scope of the task opened out. This offered more opportunity for creativity. Throughout the session, the role of driver switched between learner and facilitator with a gradual progression towards the learners working autonomously under their own direction, seeking advice rather than direction from the facilitator. This dimension is proposed to encourage the creation of learning experiences in which learners become active participants rather than passive recipients. When applied to programming in particular, a session with high *Driver Shift* offers an opportunity for learners to consolidate code comprehension with code generation [22].

3.2.6 Risk Reward

The *Risk Reward* dimension considers the relationship between the investment of effort or risk that a learner undertakes and the reward when feedback is received. Investment of effort without confirmation that the correct actions have been taken by the learner is considered a risk. This is because it may result in wasted effort or even worse, confirming an incorrect understanding or application of a skill. For a language such as Java, the amount of effort investment required from the learner to get the payback or reward of some text being displayed is considerable, so high risk. In a language like Processing, the effort investment made by the learner before observable outcome is much less; it is possible to render output in one line of code, so there is lower risk. A special case of this dimension is the time to start the first task. In Robot Dance, this was an important consideration for establishing teacher-student relationships. In all the studies conducted, the *Risk Reward* cycles were extended from initially very tight cycles of around 10 minutes per example to a larger and longer open-ended task that reflected the learners' confidence with skills being taught. One of the key decisions for the design of creative tools put forward by [13] was that they must possess a 'low floor' or enable a quick win for learners. The *Risk Reward* dimension takes this a step further than purely identifying difficulties. It encourages thought around the relationship between challenging aspects of work and the reward learners receive. An advantage of this dimension is that it encourages reflection on how much autonomy learners are given, for example arranging a looser risk reward if a learner is a 'mover' rather than a 'stopper'.

3.2.7 Grouping

The *Grouping* dimension draws attention to the different arrangements of learners that are possible. Throughout the studies conducted, three natural groupings of learners were noted: individuals, pairs, and groups of more than two people. In addition, there have been situations where there have been asymmetric groups in which learners worked with parents or with learners of different abilities, as in Robot Dance in the Community. There is a substantial body of literature exploring various approaches to group learning, including collaborative learning (e.g. [2]), team based learning (e.g. [11]), cooperative learning (e.g. [4]) and peer learning (e.g. [27]). It is beneficial for learners to experience the social complexity that working in a group brings. This needs to be balanced against the desire to support individual focus on a particular learning point. Switching groups can be a good way to reach a compromise, as achieved in Digital Makers. The duration of the session is an important consideration, however, as switching groups is potentially disruptive, which could be useful or harmful. As with the other LDs, *Grouping* highlights and provokes reflection around the merits and shortcomings of a particular design decision

3.2.8 Session Shape

A strong theme throughout [18] is the relationship between the physical environment and its affordance for better educational practices. The physical environment encapsulates all elements of the space in which learning takes place, including aspects such as the arrangement of tables and location of supporting visuals such as white boards or projectors. The physical environments involved in the studies here were classroom (Robot Dance), public space (Robot Dance in the Community), computing lab (Whack-A-Mole) and informal learning space (Digital Makers). The *Session Shape* dimension serves as a placeholder to consider what constraints and affordances are offered by the space that you inhabit with your learners. Flexibility is the most desirable attribute for a learning space. In an ideal situation, a room will have enough space to allow movement of learners as the session requires, as was seen in Digital Makers.

3.3 Working with Learning Dimensions

Learning Dimensions are intended to be a lightweight tool that can aid the design and refinement of learning experiences in programming. A web application has been written to help educators use the LDs [26]. It comprises two screens: a view screen presents information about the LDs; a notes screen gives a mechanism for educators to make relevant notes. This view screen of the application was designed to focus upon each LD. When an LD is selected, its title and brief description are presented as an aide memoire. In addition, there is a check box to indicate whether the educator has control over this aspect of the learning experience and a text box for relevant information (Figure 3). Below the description is a set of bullet points that describe different aspects of the LDs and how they may affect the learning experience being designed. This is intended to aid the educator in reflecting about how they might apply LDs to their learning experience. Underneath this description box is a text area where notes can be made. Finally, a button enables navigation to the notes screen that presents all the LD notes together. This alternative notes screen shifts the focus from the description of individual LDs to the educator's notes and those LDs over which they have control. This view allows one to see an overview of the entire learning experience and to think about how decisions relate to each other.

An example use of the application was to reflect on the design of a further workshop that allowed learners to gain an understanding of some other elementary computing concepts in a tactile learning experience using Bare Conductive Electric Paint [3]. The web application identified that three of the LDs were constrained by the nature of the task and the event, and thus could not be emphasised in the workshop design.

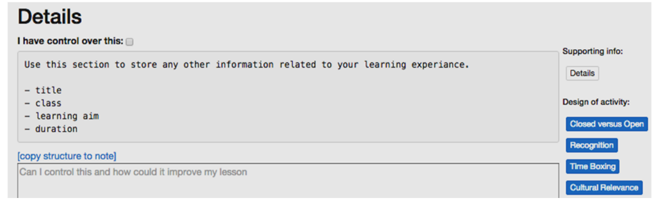

Figure 3: example edit view from web application

The remaining dimensions, in contrast, were sufficiently open that they could be tailored to influence the design of the session. Examples of notes recorded in the web application by the first author are given next.

Closed versus Open:
"Elements that were closed were chosen to support learners' lack of experience with the task. To offset the closed element, a softer open element was included to allow learners to have some control over an aspect of the learning experience.
Open elements: part of this activity involves the design of a face that incorporates flashing LEDs. This is very open as learners can design anything they wish.
Closed elements: the circuit the learners use is screen-printed and thus pre-defined. This constraint limits creativity but enables the workshop to be delivered to a wide range of learners, as the challenge is understanding a 'thing that is' rather than creating something new which is a higher order task."

Space to Play:
"The workshop naturally split into two activities that learners could perform independently, firstly designing their artwork and secondly hooking up the electronics. This aspect of *Space to Play* is valuable when encouraging a group of learners to diverge after the launch and then converge, sharing findings at the landing.

launch: introduce activity, choose which LED to include.
activity: make holes for LEDs and draw and colour picture.
landing: review and reinforce what has been created.
launch: describe how to use the conductive paint to form connection between the printed circuit and the components.
activity: hook up components.
landing: confirm it is working."

4. CONCLUSIONS

For a learning experience to be successful, it is crucial that learners are as engaged as possible. [10] described the Contributing Student Pedagogy, which aims to achieve this by enabling learners to have a prominent role in their learning experiences. CARSS is a framework for learner-centred design of educational software [8]. It offers a comprehensive set of issues associated with the design and development of educational technologies, identifying five important areas: context, roles, stakeholders, activities and skills. Creating an engaging learning experience is not a mechanical process governed by a set of rules to be followed dutifully to guarantee consistent results. It requires reflection and consideration not just of what is to be learned but also of who is learning and how they can best succeed. [17] continues to gather and share successful ideas for CS assignments and their materials. [6] likewise addresses the sharing of materials relating to assessment, also in the context of creative computing.

This paper is more general: it describes four studies of novel learning experiences which generated insights aligned with the literature and which informed the creation of a set of Learning Dimensions. The work reported here is a summary of a larger description available at [14], and here is intended to focus on things that educators can apply. The LDs have been made available to educators via a web application [26]. As a resource, they are intended to be lightweight, accessible, and easy to use. The intention is not to present a new pedagogy or theory that tackles all or even most of the aspects of the creation of programming learning experiences. Instead, the LDs are a set of important factors from which educators can select to add value and to make informed decisions about their practice. They should provoke thought about areas of opportunity in the design of an engaging learning experience and as a source of inspiration and information for educators who are critically evaluating a learning experience. Furthermore, the LDs unify published conclusions from other authors with new insights from the four field studies into a single suite. It is an inexact categorisation of some of the finer aspects of learning to program but should provide a useful background against which to assess the totality of the experience to maximise each learner's motivation and engagement.

5. ACKNOWLEDGMENTS

Nesta; University of Dundee, for funding the first study.

6. REFERENCES

[1] Arduino [Online]. Available: http://www.arduino.cc [Accessed 2014]

[2] Bagley, C.A. and Chou, C.C., 2007, June. Collaboration and the importance for novices in learning java computer programming. *ACM SIGCSE Bulletin*, 39 (3), pp. 211-215).

[3] Bare Conductive, 2014. *Creative electronic tools – Bare Conductive* [Online]. Available: http://www.bareconductive.com [Accessed 2014].

[4] Beck, L.L., Chizhik, A.W. and McElroy, A.C., 2005, February. Cooperative learning techniques in CS1: design and experimental evaluation. *ACM SIGCSE Bulletin*, 37(1), pp. 470-474.

[5] Brennan, K., Monroy-Hernández, A. and Resnick, M. 2010. Making projects, making friends: Online community as catalyst for interactive media creation. *New directions for youth development*, *2010*(128), pp.75-83.

[6] Cateté, V., Snider, E. and Barnes, T. 2016. Developing a Rubric for a Creative CS Principles Lab. *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. ACM, New York, NY, USA, pp. 290-295.

[7] Cordova, D. I. and Lepper, M. R. 1996. Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology,* 88(4), pp.715-730.

[8] Good, J. and Robertson, J. 2006. CARSS: A framework for learner-centred design with children. *International Journal of Artificial Intelligence in Education,* 16(4), pp.381-413.

[9] Green, T.R.G. and Petre, M., 1996. Usability analysis of visual programming environments: a 'cognitive dimensions' framework. *Journal of Visual Languages & Computing*, *7*(2), pp.131-174.

[10] Hamer, J., Cutts, Q., Jackova, J., Luxton-Reilly, A., McCartney, R., Purchase, H., Riedesel, C., Saeli, M., Sanders, K. and Sheard, J. 2008. Contributing learner pedagogy. *ACM SIGCSE Bulletin*, *40*(4), pp.194-212.

[11] Lasserre, P. and Szostak, C., 2011. Effects of team-based learning on a CS1 course. *ACM Conference on Innovation and Technology in Computer Science Education*. ACM, New York, NY, USA, pp. 133-137.

[12] Lego. 2010. *LEGO Mindstorms* [Online]. Available: http://mindstorms.lego.com/en-us/default.aspx [Accessed 2010].

[13] Maloney, J., Resnick, M., Rusk, N., Silverman, B. and Eastmond, E. 2010. The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*, *10*(4), p.16.

[14] Martin, C. 2017, PhD Thesis, University of Dundee.

[15] Martin, C. and Hughes, J., 2011. Robot dance: Edutainment or engaging learning. In *Proceedings of the 23rd Annual Psychology of Programming Interest Group Conference.*

[16] Nesta. 2014. *Nesta* [Online]. Available: http://www.nesta.org.uk [Accessed 2014].

[17] Nifty Assignments [Online]. Available: http://nifty.stanford.edu/ [Accessed 2016]

[18] Oblinger, D. 2006. *Learning spaces,* EDUCAUSE Washington, DC

[19] Pane, J.F. and Myers, B. A., 1996. Usability Issues in the Design of Novice Programming Systems. *Human-Computer Interaction Institute Technical Report CMU-HCII-96-101.*

[20] Perkins, D.N. and Martin, F. 1989. Fragile knowledge and neglected strategies in novice programmers. In *first workshop on empirical studies of programmers on Empirical studies of programmers* (pp. 213-229).

[21] Petre, M. and Price, B. 2004. Using robotics to motivate 'back door' learning. *Education and Information Technologies*, *9*(2), pp.147-158.

[22] Petta, P., Pelachaud, C. and Cowie, R. 2011. Emotion-Oriented Systems. *The Humaine Handbook, ISBN*, pp.978-3.

[23] Robins, A., Rountree, J. and Rountree, N. 2003. Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), pp.137-172.

[24] Sharp, H., Robinson, H. and Petre, M. 2009. The role of physical artefacts in agile software development: Two complementary perspectives. *Interactive. Computing,* 21, 108-116.

[25] Soloway, E. and Spohrer, J.C 1989. *Studying the Novice Programmer,* Hillsdale, NJ, USA, L. Erlbaum Associates

[26] http://sspog1.github.io/ LD web page [Online].

[27] Topping, K., 1998. Peer assessment between learners in colleges and universities. *Review of educational Research*, *68*(3), pp.249-276.